INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Design and Analysis of Algorithms
MA 21007/Assignment-7/Nov. 2014
Due: 12 Nov 2014

Problem 1. Running in Boston

To get in shape, you have decided to start running to work. You want a route that goes entirely uphill and then entirely downhill so that you can work up a sweat going uphill and then get a nice breeze at the end of your run as you run faster downhill. Your run will start at home and end at work and you have a map detailing the roads with m road segments (any existing road between two intersections) and n intersections. Each road segment has a positive length, and each intersection has a distinct elevation.

(a) Assuming that every road segment is either uphill or downhill, give an efficient algorithm to find the shortest route that meets your specifications.

(b) Give an efficient algorithm to solve the problem if some roads may be level (i.e., both intersections at the end of the road segments are at the same elevation) and therefore can be taken at any point.

Problem 2. Karp's minimum mean-weight cycle algorithm

Let $G = (V, E)$ be a directed graph with weight function $w : E \to \mathbb{R}$, and let $n = |V|$. We define the ***mean weight*** of a cycle $c = \langle e_1, e_2, \ldots, e_k \rangle$ of edges in E to be

$$\mu(c) = \frac{1}{k} \sum_{i=1}^{k} w(e_i) \ .$$

Let $\mu^* = \min_c \mu(c)$, where c ranges over all directed cycles in G. A cycle c for which $\mu(c) = \mu^*$ is called a ***minimum mean-weight cycle***. This problem investigates an efficient algorithm for computing μ^*.

Assume without loss of generality that every vertex $v \in V$ is reachable from a source vertex $s \in V$. Let $\delta(s, v)$ be the weight of a shortest path from s to v, and let $\delta_k(s, v)$ be the weight of a shortest path from s to v consisting of *exactly* k edges. If there is no path from s to v with exactly k edges, then $\delta_k(s, v) = \infty$.

(a) Show that if $\mu^* = 0$, then G contains no negative-weight cycles and $\delta(s, v) = \min_{0 \le k \le n-1} \delta_k(s, v)$ for all vertices $v \in V$.

(b) Show that if $\mu^* = 0$, then

$$\max_{0 \le k \le n-1} \frac{\delta_n(s, v) - \delta_k(s\ v)}{n - k} \ge 0$$

for all vertices $v \in V$. (*Hint:* Use both properties from part (a).)

(c) Let c be a 0-weight cycle, and let u and v be any two vertices on c. Suppose that the weight of the path from u to v along the cycle is x. Prove that $\delta(s,v) = \delta(s,u) + x$. (*Hint:* The weight of the path from v to u along the cycle is $-x$.)

(d) Show that if $\mu^* = 0$, then there exists a vertex v on the minimum mean-weight cycle such that

$$\max_{0 \le k \le n-1} \frac{\delta_n(s,v) - \delta_k(s\ v)}{n-k} = 0\ .$$

(*Hint:* Show that a shortest path to any vertex on the minimum mean-weight cycle can be extended along the cycle to make a shortest path to the next vertex on the cycle.)

(e) Show that if $\mu^* = 0$, then

$$\min_{v \in V} \max_{0 \le k \le n-1} \frac{\delta_n(s,v) - \delta_k(s,v)}{n-k} = 0\ .$$

(f) Show that if we add a constant t to the weight of each edge of G, then μ^* is increased by t. Use this to show that

$$\mu^* = \min_{v \in V} \max_{0 \le k \le n-1} \frac{\delta_n(s,v) - \delta_k(s\ v)}{n-k}\ .$$

(g) Give an $O(VE)$-time algorithm to compute μ^*.